Your name: Your student number:

**Department of Computer Science**
**University of Saskatchewan**
**CMPT 370-01**

30

**Midterm Examination**
**October 31, 2003**

Time Limit: 50 minutes Total Marks: 45

*This is a **closed book** exam. Please write your answers legibly in the space provided **on the examination paper**. In the discussion questions you may use point form as long as your answer is coherent. If you need more space, use the back of the page. Rough work can be done in the answer booklets. Be sure to budget your time appropriately so you can **answer all questions**. The number of marks assigned to each question is a rough guide as to the relative amount of time to spend on that question. Good luck.*

**Section 1: Short Discussion [3 marks for each question; total for the section: 15]**
*Each of the questions in this section requires a short written answer.*

1. What does it mean to "assign responsibility" for an operation?

When you assign responsibility for an operation you are specifying who, what class, must perform that operation

3

2. Why is version control important in software testing? - v.c. on data files?

It's important in case a new version fails a test. You can revert to old, functioning versions. It also helps to ensure that everyone is using the same version (CVS is a good example)

2

3. What is the difference between an interaction diagram and a sequence diagram?

An interaction diagram shows interactions between objects. These interactions are general (ie student registers in class) A sequence diagram show the order of events for a specific interaction. Sequence diagrams are specific

an s.d. is an i.d.

1

4. In what way is the "layers" architectural pattern an example of Larman's "protected variations" GRASP pattern?

The layers architecture isolates the domain layer from the services and presentation layer. This ensures that changes in either the services or presentation layer do not render the domain layer effectively useless, and require the domain layers redesign to conform to the changes.

(relate to protected variations)

2 1/2

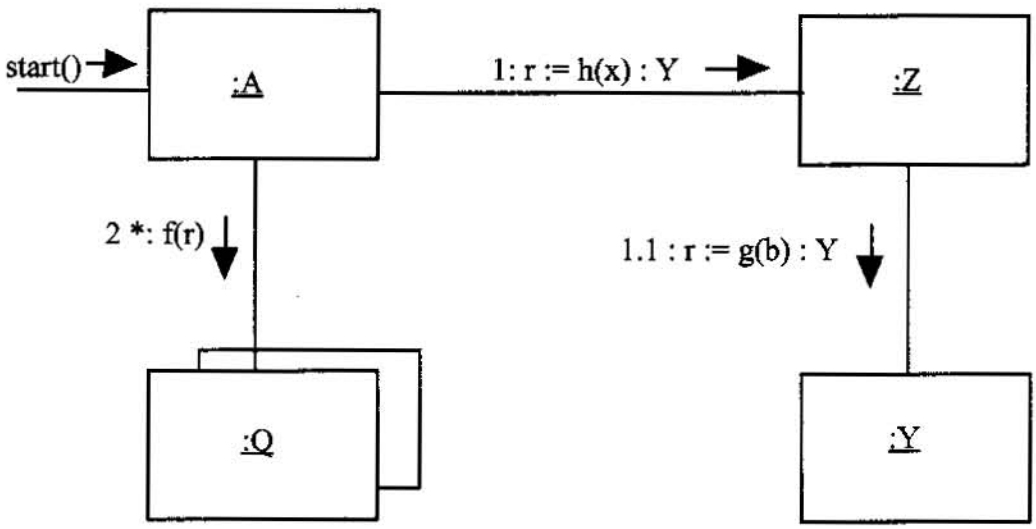
5. Why does a class diagram object have three parts and a domain model object have only two?

The domain model is a conceptual representation of the Real world, and therefore cannot have software functions or methods. A class diagram is a mapping of the Real world into the software world, and as such can contain both attributes and functions or methods

3

**Section 2: Visibility [total for the section: 6]**

*Consider the following collaboration diagram and answer the questions below.*




(a) Write down the sequence in which these objects are invoked. [2 marks]

They are invoked in the following order:
A then Z then Y then Q *repeated

repeated

1½/2

(b) For each object indicate which other objects are *visible* to that object and what kind of visibility it is. [4 marks]

parameter
local
global
attribute

A) Z is attribute visible to A.
Y is parameter visible to A also locally visible
Q is locally visible to A

Z) Y is attribute visible to Z

Q) Y is parameter visible to Q

Y) ??

need to specify nothing is visible to Y

3

**Section 3 – Analysis and Design [marks for each part indicated; total for the section: 24]**
*This question has several parts. Do your best to answer each part in the space available.*

The Just-Out-Of-Bankruptcy airline (AirJOOB) has hired you to help to create a new automated system for them. This system is to be used by agents in AirJOOB's call centre to carry out activities on behalf of customers who phone in. The overall goals of the system are ambitious: to provide information to the agents about flights, to allow the agents to reserve flights for their customers, to allow agents to retrieve information about possible destinations (eg. hotel information, recommendations about things to do, weather and climate information, etc.).

(a) Describe, in one sentence each, the important use cases for this system. Be sure to indicate which actors are involved in each such use case as well as what the use case does, overall. [3 marks] • Get flight Information is done by the agent (actor). The agent requests information about one or more flights, and the system returns said information

3

•Reserve Flight is done by the agent (user). A Customer telephones the AirJOOB's call centre, and an agent answers. The customer then requests to book a flight. The agent gets the flight information and reserves one to many seats on that flight in the customer's name.

•Get destination information is performed by the agent (user) A customer requests information on his or her destination. The agent then requests that information from the system. The requested information is displayed (outputted) for the agent.

(b) In planning the overall project during the inception phase, in what order would you schedule the use cases to be designed and implemented? Why? [2 marks]

I would implement the Get destination use case last as it is the least essential to the running of an airline based upon other expert pattern. I would implement the Reserve flight use case first, as it is most likely the most complicated, and that follows unified process design principles.

2

(c) Identify by name the major software classes that you would create in the overall system. Only the names need be provided: you do *not* need to provide a class concept diagram (or a domain model). [4 marks] • Flight • Destination • Agent

• Customer • Airport • Plane •

why ?

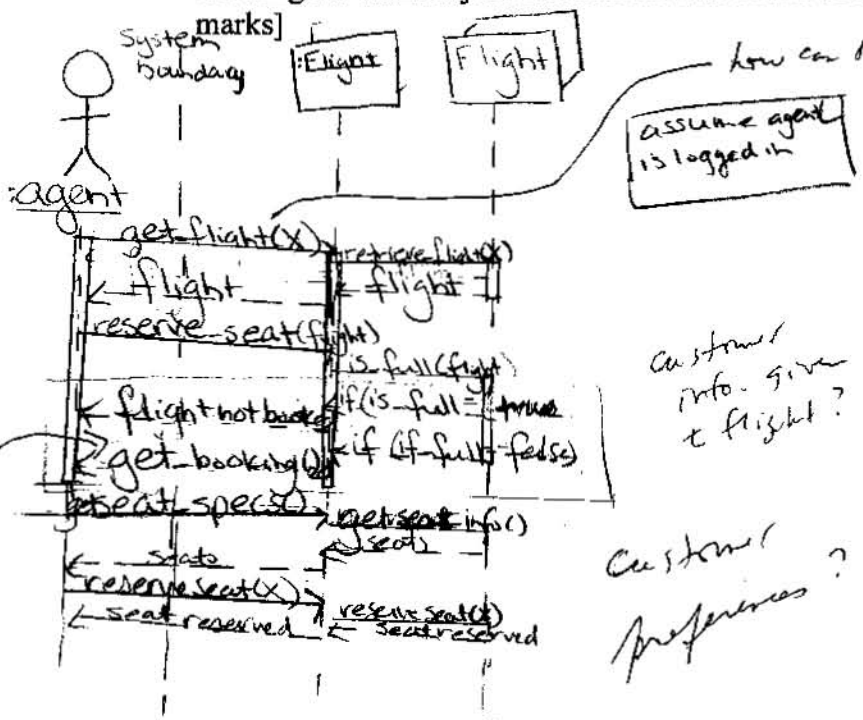
payments ?

specific destination info. ? → hotel, activities, climate

AirJOOB's call centre ?

2

(d) Assume there is an operation in one of the use cases called **bookFlight** that actually reserves a seat on a particular flight for a customer. Draw a sequence diagram or a collaboration diagram showing the main object interactions that would be undertaken to achieve **bookFlight**. [10 marks]




How can this be sent to an instance of flight when you haven't yet retrieved the flight from the container class?

- this also needs to get customer info for seat booking
- get payment authorization + price in another use case / operation

customer info. given to flight?

customer preferences?

State other assumptions customer exists, flight exists, could assume flight not full

I don't understand why the container does is doing all this work?

- I put the conditional in a box indicating the course of events for success/failure.
- when a flight_not_booked message appears the agent will get flight information to find a new flight

5

(e) For the methods invoked in your diagram in (d) what design patterns did you choose and why? [5 marks]

other design patterns?

- retrieve_flight(X) by expert because flight knows (? (see my remarks above)

but you're moving the flights container class do all of this?

- reserve_seat(X) by expert because flight contains information on how many seats are left
- is_full(X) by expert because flight knows if all of it's seats are filled
- get_booking(X) expert. reasoning as above

2

***Beware of the Goblins Tonight!***

- getSeat_specs() expert reasoning as above.
- reserve_seat(X) controller 4 of 4 + expert the flight controls which seats can be booked, it has this information